第 37 卷第 1 期 计算机应用研究 Vol. 37 No. 1
录用定稿 Application Research of Computers Accepted Paper

一种基于 MapReduce 并行化计算的大数据聚类算法 *

张文杰[1,2]，蒋烈辉[1,2]

(1. 解放军信息工程大学 网络空间安全学院，郑州 450001; 2. 数字工程与先进计算国家重点实验室，郑州 450001)

摘　要：面对大数据规模庞大且计算复杂等问题，基于 MapReduce 框架采用两阶段渐进式的聚类思想，提出了改进的 K-means 并行化计算的大数据聚类方法。第一阶段，该算法通过 Canopy 算法初始化划分聚类中心，从而迅速获取粗精度的聚类中心点；第二阶段，基于 MapReduce 框架提出了并行化计算方案，使每个数据点围绕其邻近的 Canopy 中心进行细化的聚类或合并，从而对大数据实现快速、准确地聚类分析。在 MapReduce 并行框架上进行算法验证，实验结果表明，所提算法能够有效地提升并行计算效率，减少计算时间，并提升大数据的聚类精度。

关键词：大数据；MapReduce；并行计算；数据聚类

中图分类号：TP391　　doi: 10.19734/j.issn.1001-3695.2018.05.0496

Parallel computation algorithm for big data clustering based on MapReduce

Zhang Wenjie[1, 2], Jiang Liehui[1, 2]

(1. *Faculty of Cyberspace Security*, *PLA Information Engineering University*, *Zhengzhou* 450001, *China*; 2. *State Key Lab Math Engn & Adv Comp*, *Zhengzhou* 450001, *China*)

Abstract: Aiming at solving the problem of big data's large scale and complex computation, this paper adopt the idea of two-stage progressive clustering, and proposes a parallel computation algorithm for big data clustering based on MapReduce. In the first stage, our method acquires the initialized clustering center through Canopy algorithm, in order to find relatively accurate cluster center points quickly. In the second stage, , we present a novel scheme of parallel computation based on MapReduce framework, which makes each data node cluster or merge around its adjacent Canopy center node. In this way, the algorithm can make the procedure of data clustering fast and accurately. The results of the experiments deployed on MapReduce show that our algorithm can effectively improve the efficiency of parallel computing, reduce computing time, and improve big data's clustering accuracy.

Key words: big data; MapReduce; parallel computation; data clustering

0　引言

随着互联网通信、数据存储、信息处理等技术的快速发展，各行各业都需要存储、分析和处理爆炸性增长的业务数据。网络大数据的分析与处理，已经发展成为当前非常重要的研究领域。大数据体量庞大且计算复杂等问题，严重限制了信息产业的技术应用于发展[1,2]。在大数据背景下如何实现快速高效的数据聚类，已经成为大数据挖掘与分析亟需解决的重要问题[3,4]。

聚类分析是大数据分析领域一个重要的研究方向，K-means 算法是其中应用较为广泛的经典算法之一。该算法具有应用简便、聚类速度快等特点[5]，但需要预设置 K 值（聚类中心数量），随机选择聚类中心点可能导致聚类结果为局部最优。在大数据条件下，K-means 算法的这些问题进一步凸显，影响大数据聚类的精度和效率。针对这些问题，近年来有相关的研究工作开始展开。文献[6]提出了一种基于 Hadoop 平台的 K-means 并行计算的聚类方法，该算法通过引导大数据集聚类中心的初始划分，减小随机选择聚类中心的不确定性，并基于 Hadoop 平台提出了并行化计算方法，以改善大数据聚类的效率和准确性。文献[7]提出一种并行化的 PAM 聚类算法，该算法结合蚁群算法以提升聚类迭代过程中的搜索性能，从而提升 PAM 聚类算法的收敛速度，并改善算法的大数据聚类性能。文献[8]在 MapReduce 框架上，提出一种基于样本预处理策略的 K-means 并行算法。该算法结合 K 选择排序算法进行并行采样，以提高采样效率，提升聚类算法的运行效率。

针对 K-means 聚类算法在大数据环境下效率不高等问题，本文基于 MapReduce 框架，提出了改进的 K-means 并行化计算的大数据聚类方法。区别于传统的聚类算法，该算法采用了两阶段渐进式的聚类思想。在第一阶段，通过 Canopy 算法初始

收稿日期：2018-05-06；**修回日期：**2018-06-27　**基金项目：**河南省基础前沿课题（142300410090）；河南省科技攻关计划项目（162102210035）
作者简介：张文杰（1978-），男，河南郑州人，工程师，硕士，主要研究方向为大数据技术及其应用（zhaifei651@163.com）；蒋烈辉（1967-），男，河南郑州人，教授，博士，主要研究方向为计算机体系结构、大数据技术及其应用.

化划分聚类中心，从而迅速获取粗精度的聚类中心点；在第二阶段，基于 MapReduce 框架给出了并行化设计方案，使外围数据点围绕其邻近的 Canopy 中心完成进一步地细化聚类或合并，将每个数据点划分至距离最近的聚类中心，并重新计算每个聚类中心新的中心点，从而对大数据实现快速、准确地聚类分析。

引入 Canopy 算法，每次只比较落在同一区域内对象与中心点之间的距离，通过减少比较次数大大降低整个聚类的运行时间，从而提高算法的计算效率，优化大数据的聚类过程。更重要的是，本文算法通过引入“最小最大原则”，避免了人工设置区域半径 T1、T2 带来的干扰，使得任意两个 Canopy 中心点之间的距离尽可能远，从而避免聚类过程陷入局部最优。

1　研究背景概述

2004 年，Google 公司推出了 MapReduce 编程模型以批量处理大数据集，并基于此开发出了 Hadoop 大数据批量处理架构。Hadoop 是一个开源的高效云计算基础架构平台，利用通用的硬件就可以构建一个强大、稳定、简单，并且高效的分布式集群计算系统。

MapReduce 提供了一种新的对海量数据的处理方式，通过抽象出分层次的编程模型，从而大大简化将大数据分片成子任务，并同时在集群计算机上运行的过程[9,10]。

MapReduce 框架一般将大数据并行计算划分为 Map、Combine、Reduce 三个步骤。在 Map 阶段，Map 函数将输入数据转换为<key, value>序列；在 Combine 阶段，Combine 函数对 Map 函数的输出结果在本地进行合并和处理，以减小大数据聚类过程中的 I/O 负担；在 Reduce 阶段，Reduce 函数将获得的<key, value>序列按照算法设定的规则进行聚类处理。MapReduce 并行计算的框架如图 1 所示。通过利用 MapReduce 框架和接口，能够简化并行化开发过程，便于有效地组织和应用分布式资源，高效便捷地进行大数据分析和计算。

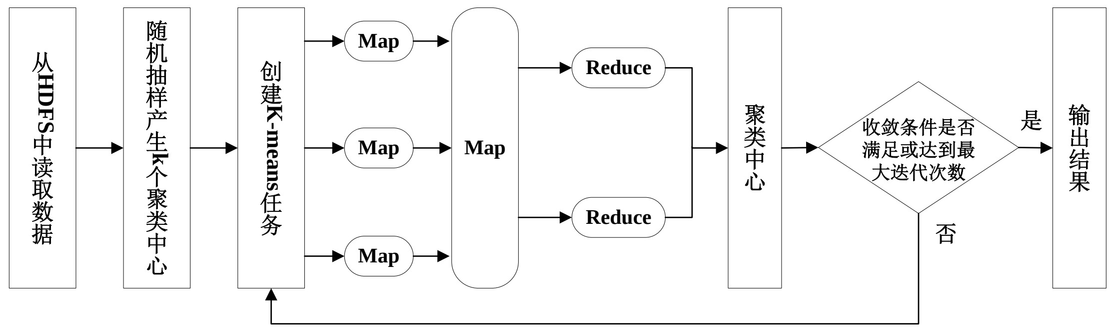


图 1　基于 MapReduce 的 K-means 并行算法框架

Fig.1　K-means parallel algorithm framework based on MapReduce

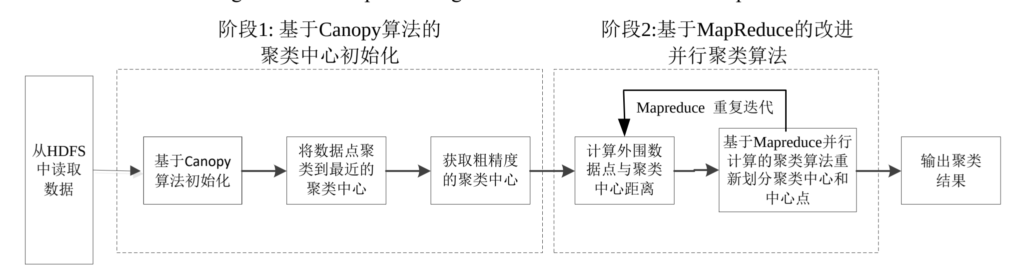


图 2　基于 MapReduce 的并行化聚类算法流程

Fig.2　Parallel clustering algorithm flow based on MapReduce

2　基于 MapReduce 的并行化聚类算法

基于 MapReduce 的并行化聚类算法流程如图 2 所示。

K-means 算法一般需要随机设置初始化的聚类中心点，使得聚类结果易受中心点选取的影响，造成聚类结果往往出现局部最优，甚至不稳定[11]。针对这些问题，本文提出一种两阶段渐进式的聚类算法。该算法在第一阶段采用 Canopy 算法获取初始化的聚类中心，从而迅速获取粗精度的聚类中心点；第二阶段基于 MapReduce 框架采用并行化计算的改进设计，迭代计算外围数据点与其邻近聚类中心的距离，将每个数据点划分至距离最近的聚类中心，并重新计算每个聚类中心新的中心点。改进后的 K-means 算法通过两阶段渐进式的聚类，通过局部聚类中心和中心点的调整，实现对大数据快速、准确地聚类分析，从而大大减少计算量和复杂度，同时避免了聚类结果陷入局部最优效果的问题，有效提升了算法的整体聚类精度。

2.1　初始化聚类中心

Canopy 算法计算速度快，但聚类精度较粗，适合用于获取大数据的初始化聚类中心[12,13]。该算法需要设置两个距离阈值 T_1、T_2（$T_1 < T_2$），初始化聚类中心的具体流程（图 3）如下：

a)将大数据集中的数据点存储为 List 集合；

b)从 List 中随机删除一个点 P，这个点构成一个新的聚类中心 Clu_p；

c)搜索 List 中剩下的数据点，若某数据点 Q 与点 P 的间距小于 T_2，则将点 Q 并入到聚类中心 Clu_p 中；

d)遍历聚类中心 Clu_p 中所有的数据点，若数据点 i 与点 P 的间距小于 T_1，则将点 i 从 List 集合中删除；

e)重复执行步骤 b) ~d)，将集合 List 中所有数据点都划分到对应的聚类中心；

f)停止迭代，获取大数据集的初始化聚类中心。

Canopy 初始中心点的个数决定了聚类类别数 K，该值一般由经验或者多次实验设定。为解决 Canopy 区域半径 $T1$、$T2$ 以及初始中心点的随机选取问题，本文引入“最小最大原则”对该算法进行改进，提出 Canopy 中心点的优化选取与设置方法。该方法的基本思路是：将数据集合划分为若干个 Canopy，任意两个 Canopy 中心点之间的距离代表聚类类别间距。为了避免聚类过程陷入局部最优，初始的 Canopy 中心点间距应尽可能远。

Canopy 算法能够以较小的代价，粗略地将大数据集划分成若干个重叠子集，每个子集可视为一个聚类中心。通过 Canopy 算法对数据点进行粗略划分，然后在此基础上进行进一步的聚类。而且通过聚类中心划分，每个数据点已经明确其隶属关系，从而无须计算每个点与每个中心点的距离，而只需计算该点和其所在中心点的距离，并将其归属到距离最近的聚类中心即可，从而大大减少了计算量和复杂度。

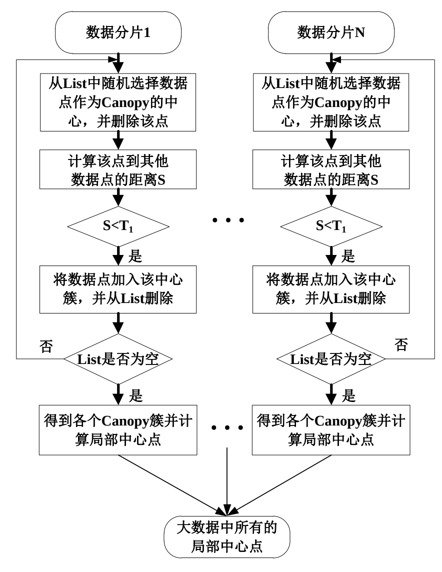


图 3 基于 Canopy 算法获取初始化聚类中心的流程

Fig.3 Obtaining initial clustering center based on Canopy algorithm

2.2 聚类算法流程

改进后的 K-means 算法首先通过 Canopy 算法获得初始化的聚类中心，然后基于初始化聚类中心进行 K-means 聚类迭代，并最终获得聚类结果。聚类流程如下：

a)获取初始化聚类中心。对于给定的大数据集 X，通过 Canopy 算法设定的距离阈值 T_1、T_2（$T_1 < T_2$），划分初始化的聚类中心。

b)数据点划分。经 Canopy 算法划分聚类中心后，每个数据点至少属于一个聚类中心或多个聚类中心。数据点不需要再计算该点到各中心点的距离，而只需计算并比较该点所属聚类中心点的距离，并将该点划分到距其最近的聚类中心。重新划分数据点后，各个聚类中心之间不会有重叠的数据点。

c)更新 K-means 中心点。获取重新划分的聚类中心后，采用 K-means 求平均的方法，计算并获取各个聚类中心的新中心点。

d)合并 K-means 中心。比较各中心点之间的距离，获取距离较近的中心点及对应的聚类中心。

e)合并聚类中心。合并聚类较近的聚类中心，并计算更新合并后的聚类中心点。采用 Canopy 算法思想，重复步骤 a)获取新的重叠聚类中心。重复步骤 b)~d)，对各聚类中心数据点进行聚类迭代，直至算法收敛。

f)形成聚类结果。在算法收敛后，将大数据集中各个数据点划分到 K 个聚类中心，形成 K 个不重叠的聚类子集，完成大数据聚类。

聚类过程中不仅要对数据点进行吸纳划分，对数据点所归属的聚类中心也要进行相应合并。合并后缠身新的聚类中心，需要重新计算聚类中心的中心点，并计算各数据点至该中心点的距离，从而反复聚类迭代，最终实现对整个大数据集的有效聚类划分。

2.3 基于 MapReduce 改进的聚类算法的并行化设计（图 4）

为了提升对大数据的处理能力，进一步提高聚类效率和可扩展性，基于 MapReduce 框架分别在 Map、Combine、Reduce 这三个阶段实现本文算法，并行化聚类大规模数据。

本文选择 Hadoop 平台中的 MapReduce 框架完成并行化计算对的聚类过程。首先，基于初始化策略，从存储在 HDFS 的输入数据集中选取 K 个数据点作为初始聚类中心点；然后，执行 K-means 改进算法的并行化计算，将计算任务分解为 Map、Combine 和 Reduce 函数，并在并行化迭代计算过程中完成大数据聚类。

1）Map 阶段

在该阶段中，Map 任务接收序列文件中的每一行作为不同的键值对＜key,value＞，并形成 Map 函数的输入。首先，Map 函数计算每个数据点与各聚类中心点之间的距离；其次，Map 函数根据距离最短原则，将每个数据点划分与其距离最近的聚类中心；最后，Map 函数输出中间数据至 Combine 函数。

2）Combine 阶段

在该阶段中，首先，Combine 函数从 Map 函数输出的 value 中提取所有的数据点，并合并属于相同聚类中心的数据点；其次，Combine 函数统计分配在统一聚类中心的数据点的个数，并计算数据点的均值；最后，Combine 函数将数据排序、重组、分片，并输出每个中心的局部聚类结果至 Reduce 函数。

3）Reduce 阶段

在该阶段中，首先，Reduce 函数从 Combine 函数输出的 value 中提取所有的数据点，并聚合所有聚类中心的局部结果；其次，Reduce 函数为每个聚类中心计算新的聚类中心点；最后，Reduce 函数判断准则函数是否收敛。若准则函数已收敛，

Reduce 函数将输出最终结果，否则将执行下一次迭代。

本文算法的计算任务主要包括下三个方面：a）计算数据点与聚类中心之间的距离；b）将每个数据点划分至距离最近的聚类中心；c）重新计算每个聚类中心新的中心点。其中，Map 函数产生局部数据集的聚类中心，遍历子集中的所有数据点，判断其与聚类中心的距离；Reduce 阶段收集局部中心点，并执行与 Map 阶段同样的操作，合并局部的聚类中心，形成新的聚类中心和中心点，并以<中心点 ID，中心点各维度值>形式存放在本地。最终，本文算法将局部数据点归到与其距离最近的 K 中心，输出<数据点 ID：数据各维值，所属中心>，从而获得聚类结果。

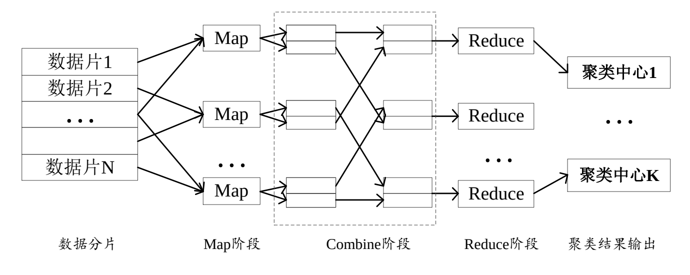


图 4　基于 MapReduce 改进的聚类算法的并行化设计

Fig.4　Parallel design of improved clustering algorithm based on MapReduce

3　实验结果与分析

实验选择四台 Lenovo 台式机构建 MapReduce 平台，配置为 Intel(R) 6 Core(TM) i3-3240 3.39 GHz CPU，4.0 GB 内存，Ubuntu 14.04 的操作系统。选择其中三台电脑为 datanode，剩下一台为 namenode。实验主要采用了三个典型数据集。一个是 UCI 数据库[14]中 Iris 数据集，其中包含 150 个数据样本，每个样本包括四个维度。Iris 数据集的分类已知，能够用于评估聚类算法的准确性。第二个是德国手机定位数据集 Crowdflow，该数据集为 1.01 GB，包含 13 082 242 定位数据。第三个是 MIT 林肯实验室收集了 9 周的 TCPdump 网络连接数据，形成的 KDD CUP99 数据集。其 10%的训练数据集包含 23 种攻击类型，494 021 条记录，每条记录包含 41 个特征属性和 1 个类别属性，各属性之间用逗号分隔。由于该数据集规模庞大、结构复杂，且关于聚类特征的先验信息很充分，非常适合作为大数据聚类的测试数据。这三个大数据集的每条数据都包含精度、维度、水平精度等多维属性，属于典型的大数据，能够有效验证和评估算法的聚类速度、扩展性、精度等性能。

将本文算法与 K-means、CTK 和 MMKMEANS 这三种典型算法进行比较，并统一在 Hadoop 平台上进行聚类对比。首先需要根据各个数据集的实际情况设置参数：由于 Iris 数据集共包含三类数据，设置 K=3；德国手机定位数据主要围绕城市地理分布形成了 63 个聚类中心，设置 K=63。

表 1 所示为在 Iris 数据集上四种算法的聚类结果。由于 Iris 数据集较小，算法运行时间区别较小，主要比较数据聚类的准确率(precision)。相比传统的 K-means 算法，本文算法将正确聚类个数从 132 提升到 143，准确率提升了 7.3%。相比其他两种算法，本文算法的数据点划分更为合理，聚类结果也最接近 Iris 数据的实际分类，聚类精度在同类算法中是最高的。在迭代次数方面，本文算法由于需要初始化聚类中心，增加了相应的迭代次数和计算量；但是在小规模数据条件下，迭代次数增加量十分有限，增加的并行化计算时间也几乎可忽略不计。

Crowdflow 数据集中数据点较多，数据维度也更为复杂，适合比较在 MapReduce 框架下的并行计算性能。表 2 所示为四种算法在 Crowdflow 数据集上的聚类结果。KDD CUP99 数据集中特征属性的维度更多，结构更加复杂，因此数据聚类的计算时间更长，聚类准确度相对较高。表 3 所示为四种算法在 KDD CUP99 数据集上的聚类结果。综合四种算法在两个大数据集上的聚类性能对比可以发现，在运行时间上，本文算法消耗时间最少，K-means 算法需要的运行时间最多，说明在大数据条件下，本文算法尽管在增加了初始化聚类中心的计算过程，但该算法优化了聚类计算流程和并行化计算流程，使得本文算法在大数据条件下运行时间大大减少，计算效率大大提高。同时，相比其他算法，本文算法在聚类准确率和查全率方面都有了不同程度的提升，说明两阶段渐进式的聚类思想更适用于大数据聚类，能够有效地减少运行时间，提升数据聚类的准确率和查全率。

表 1　四种算法在 Iris 数据集上的聚类性能对比

Table 1　Comparison of clustering performance between four algorithms on Iris dataset

聚类算法	聚类个数分布	Precision/%	迭代次数
本文算法	(43,57,50)	95.3	9
K-means	(32,50,68)	88.0	6
CTK	(37,54,59)	91.2	8
MMKMEANS	(39,55,56)	89.8	8

表 2　四种算法在 Crowdflow 数据集上的聚类性能对比

Table 2　Comparison of clustering performance between four algorithms on Crowdflow dataset

聚类算法	运行时间/ms	准确率/%	查全率/%
本文算法	83784	85.4	82.6
K-means	153568	63.5	60.8
CTK	134750	75.2	73.1
MMKMEANS	122943	72.9	68.5

表 3　四种算法在 KDD CUP99 数据集上的聚类性能对比

Table 3　Comparison of clustering performance between four algorithms on KDD CUP99 dataset

聚类算法	运行时间/ms	准确率/%	查全率/%
本文算法	98656	88.2	83.9
K-means	180250	67.4	65.8
CTK	141593	79.5	76.4
MMKMEANS	132718	83.3	71.5



4　结束语

针对传统的 K-means 算法在大数据条件下聚类存在的问题，本文提出一种两阶段渐进式的聚类算法。该算法在第一阶段采用 Canopy 算法获取初始化的聚类中心，从而迅速获取粗精度的聚类中心点；第二阶段基于 MapReduce 框架给出了并行化设计方案，使外围数据点围绕其邻近的聚类中心完成进一步地细化聚类或合并，并重新计算每个聚类中心新的中心点，从而对大数据实现快速、准确地聚类分析。通过在两个不同类型和大小的数据集进行实验，发现改进的 Kmeans 算法通过两阶段渐进式的聚类，大大减少计算量和复杂度，同时避免了聚类结果陷入局部最优效果的问题，有效提升了算法的整体聚类精度。

本文以海量的互联网定位数据聚类、教育数据等作为应用背景，利用 MapReduce 的并行计算框架，将 Canopy-Kmeans 算法进行并行扩展。本文算法主要基于距离指标进行聚类，对于语义分析、行为模式等复杂场景的应用还不太适应。在下一步研究中，需要对相关算法进行扩展，使大数据聚类向着更加智能、有效和自适应的方向发展。

参考文献：

[1] 程学旗，靳小龙，王元卓，等. 大数据系统和分析技术综述 [J]. 软件学报，2014，25 (9): 1889-1908. (Cheng Xueqi, Jin Xiaolong, Wang Yuanzhuo, etal. Survey on big data system and analytic technology [J]. Journal of Software, 2014, 25 (9): 1889-1908.)

[2] 郭平，王可，罗阿理，等. 大数据分析中的计算智能研究现状与展望 [J]. 软件学报, 2015, 26 (11): 3010-3025. (Guo Ping, Wang Ke, Luo Ali, *et al.* Computational intelligence for big data analysis: current status and future prospect [J]. Journal of Software, 2015, 26 (11): 3010-3025.)

[3] 李馨. 高等教育大数据分析: 机遇与挑战 [J]. 开放教育研究, 2016, 22 (4): 50-56. (Li Xin. Big Data analytics in higher education: opportunities and challenges [J]. Open Education Research, 2016, 22 (4): 50-56.)

[4] 杜治娟，王硕，王秋月，等. 社会媒体大数据分析研究综述 [J]. 计算机科学与探索, 2017, 11 (1): 1-23. (Du Zhijuan, Wang Shuo, Wang Qiuyue, *et al.* Survey on social media big data analytics [J]. Journal of Frontiers of Computer Science and Technology, 2017, 11 (1): 1-23.)

[5] 宋建林. K-means 聚类算法的改进研究 [D]. 合肥: 安徽大学, 2016. (Song Jianlin. Research on improvement of K-means clustering algorithm [D]. Hefei: Anhui University, 2016.)

[6] 李晓瑜，俞丽颖，雷航，等. 一种 K-means 改进算法的并行化实现与应用 [J]. 电子科技大学学报, 2017, 46 (1): 61-68. (Li Xiaoyu, Yu Liying, Lei Hang, etal. The parallel implementation and application of an improved K-means algorithm [J]. Journal of University of Electronic Science and Technology of China, 2017, 46 (1): 61-68.)

[7] 赵宝文，徐华. 基于 MapReduce 的并行 MRACO-PAM 聚类算法 [J]. 计算机工程与科学, 2017, 39 (10): 1801-1806. (Zhao Baowen, Xu Hua. A parallel MRACO-PAM clustering algorithm based on MapReduce [J]. Computer Engineering & Science, 2017, 39 (10): 1801-1806.)

[8] 王永贵，崔鹏. 一种基于 MapReduce 高效 K-means 并行算法 [J]. 辽宁工程技术大学学报：自然科学版, 2017, 36 (11): 1204-1211. (Wang Yonggui, Cui Peng. An efficient K-means parallel algorithm based on MapReduce [J]. Journal of Liaoning Technical University: Natural Science Edition, 2017, 36 (11): 1204-1211.)

[9] 陈爱平. 基于 Hadoop 的聚类算法并行化分析及应用研究 [D]. 成都: 电子科技大学, 2015. (Chen Aiping. Research on parallelization analysis and application of clustering algorithm based on Hadoop [D]. Chengdu: University of Electronic Science and Technology of China, 2015.)

[10] 夏大文. 基于 MapReduce 的移动轨迹大数据挖掘方法与应用研究 [D]. 重庆: 西南大学, 2016. (Xia Dawen. MapReduce-based methodologies of mobile trajectory big data mining and its application [D]. Chongqing: Southwest University, 2016.)

[11] 宋旭东，朱文辉，邱占芝. 大数据 K-Means 聚类挖掘优化算法 [J]. 大连交通大学学报, 2015, 36 (3): 91-94. (Song Xudong, Zhu Wenhui, Qiu Zhangzhi. Big data K-Means clustering mining optimization algorithm [J]. Journal of Dalian Jiaotong University, 2015, 36 (3): 91-94.)

[12] 樊同科. 云环境下基于 MapReduce 的用户聚类研究与实现 [J]. 电子设计工程, 2016, 24 (10): 35-37. (Fan Tongke. Research and implementation of user clustering based on Map Reduce in cloud environment [J]. International Electronic Elements, 2016, 24 (10): 35-37.)

[13] 张友海，李锋刚. 基于 MapReduce 的 Canopy-Kmeans 算法的并行化 [J]. 辽宁科技学院学报, 2017, 19 (1): 4-5. (Zhang Youhai, Li Fenggang. Parallelized canopy-K-means algorithm based on MapReduce [J]. Journal of Liaoning Institute of Science and Technology, 2017, 19 (1): 4-5.)

[14] Murphy P M, Aha D W. UCI repository of machine learning database [DB/OL]. (2006-05-12) . http://www. ics. uci. edu/mlearn/MLReposi-tory. html.